

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

5
6

8
9
10

14

16
17
18

19

20
2122
23

24

25

1 The current Office Action rejected claims 2-9, 12, 20-22, 31, 34-47, 51, and
2 57 under 35 U.S.C. §103(a).

3
4
5
6 RESPONSE TO NOTATION REGARDING INFORMATION DISCLOSURE STATEMENT

7
8 It is believed that a legible copy of each item listed for the 21 March 2006
9 IDS was included in the original submission. Nevertheless, the PTO-1449 that
10 was submitted on 21 March 2006 and another legible copy of each item listed
11 thereon accompany this Reply.

12
13
14
15 RESPONSE TO CLAIM OBJECTIONS

16
17 The objected to claims 40 and 53 have been amended in manners believed
18 to overcome the instituted objections.

19
20 Accordingly, withdrawal of the claim objections is hereby respectfully
21 requested.

1 RESPONSE TO REJECTIONS UNDER 35 U.S.C. § 112

2
3 The current Office Action rejected claims 2-4, 6-9, 20-22, 29-31, 36-39,
4 and 48-59 under 35 U.S.C. §112, second paragraph.
5

6 The asserted indefiniteness “for failing to particularly point out and
7 distinctly claim the subject matter which applicant regards as the invention” is
8 hereby respectfully traversed generally. Nevertheless, to facilitate prosecution of
9 the instant Patent Application and to expedite its ultimate allowance as a U.S.
10 Patent, certain claims have been amended in manners believed to overcome the
11 instituted 35 U.S.C. §112, second paragraph, rejections.
12

13 Moreover, the rejections with respect to claims 29, 30, 31, 48, and 53 are
14 specifically traversed. With regard to claims 29, 30, and 31, it is respectfully
15 submitted that the “relative” terminology is internally linked so as to make it
16 definite. One of ordinary skill in the art, in conjunction with the disclosure of the
17 instant Patent Application, would know what is meant by “level of abstraction” and
18 would understand which of two layers is relatively lower and which is relatively
19 higher. With regard to claim 48, it is respectfully submitted that the allegedly
20 “relative” terminology is internally linked so as to make “simpler”, “increasingly
21 complex”, “progression”, etc. definite within the claim. Similarly, with regard to
22 claim 53, “simplicity” as a concept that is considered in an evaluation would be
23 understood by one of ordinary skill in the art, at least in conjunction with the
24 disclosure of the instant Patent Application.
25

1 Accordingly, withdrawal of the 35 U.S.C. §112 rejections is hereby
2 respectfully requested.

3
4
5
6 RESPONSE TO REJECTIONS UNDER 35 U.S.C. § 101

7
8 The current Office Action rejected claims 1-33 and 48-59 under 35 U.S.C.
9 §101. The propriety and legitimacy of the 35 U.S.C. §101 rejections is hereby
10 respectfully traversed. Nevertheless, to facilitate prosecution of the instant Patent
11 Application and to expedite its ultimate allowance as a U.S. Patent, certain claims
12 have been amended in manners believed to overcome the instituted 35 U.S.C. §101
13 rejections.

14
15 Claims 1, 15, and 48 have been amended to recite: **realizing the derived**
16 **API in one or more processor-accessible storage media.**

17 Claim 49 has been amended to address if the condition is met or is not met
18 and to recite: that certain components/types are **realized in one or more**
19 **processor-accessible storage media.**

20
21 Accordingly, withdrawal of the 35 U.S.C. §101 rejections is hereby
22 respectfully requested.
23
24
25

1 ARGUMENT(S) AGAINST THE 35 U.S.C. §§ 102(b) AND 103(a) REJECTIONS
2 AND ARGUMENTS FOR CLAIM PATENTABILITY
3

4 Generally, the current Office Action rejected claims 1, 10, 11, 13-19, 23-30,
5 32, 33, 48-50, 52-56, 58, and 59 under 35 U.S.C. §102(b) and rejected claims 2-9,
6 12, 20-22, 31, 34-47, 51, and 57 under 35 U.S.C. §103(a).

7 Specifically, the current Office Action reads as follows at the noted
8 paragraph numbers (with any highlighting removed in this Reply):

9 11. Claims 1, 10, 11, 13-19, 23-30, 32, 33, 48-50, 52-56, 58, and 59
10 are rejected under 35 U.S.C. 102(b) as being anticipated by Burger et al. (US
11 5,097,533).

12 13. Claims 2-4 are rejected under 35 U.S.C. 103(a) as being
13 unpatentable over Burger et al. (US 5,097,533) in view of Hayes (US 6,006,279).

14 14. Claims 5-9, 12, 20-22, 31, 34, 35, 37-45, and 47 are rejected
15 under 35 U.S.C. 103(a) as being unpatentable over Burger et al. (US 5,097,533)
16 in view of Corrie, Jr. et al. (US 5,495,571).

17 15. Claim 36 is rejected under 35 U.S.C. 103(a) as being
18 unpatentable over Burger et al. (US 5,097,533) in view of Corrie, Jr. et al. (US
19 5,495,571) as applied to Claim 34 above, and further in view of Hayes (US
20 6,006,279).

21 16. Claims 46, 51, and 57 are rejected under 35 U.S.C. 103(a) as
22 being unpatentable over Burger et al. (US 5,097,533).

23
24 The current Office Action therefore rejects the claims using one or more of
25 Burger et al., Hayes, and Corrie, Jr. et al.

1 Burger et al. is apparently directed to a "System and method for interfacing
2 computer application programs written in different languages to a software
3 system" (the Title). The Abstract of Burger et al. reads as follows:

4 A support system and method for interfacing of computer application
5 programs written in a plurality of languages to a software system such as a
6 database manager of the like. A plurality of generic application program
7 interfaces or entry points are defined having a corresponding plurality of
8 parameters in a consistent form required by the system to execute functions. The
9 parameters are transformations of like parameters associated with the application
10 programs which call the APIs. Processor states corresponding to threads in the
11 application programs are stored in a table shared by the generic APIs. Upon
12 return from the call and execution of the system function, processor state is
13 restored and control returned to the application program. Necessity for separate
14 entry points for applications written in each different supported language is
15 thereby avoided as well as associated increased development effort,
16 maintenance, and support.

17 Hayes is apparently directed to a "Plug-in module host framework" (the
18 Title). The Abstract of Hayes reads as follows:

19 Computer-executable process steps stored on a computer-readable
20 medium to provide an application programming interface (API) to a client
21 application for communicating between the client application and a plug-in
22 module. The API includes a menu-create routine which searches for plug-in
23 modules, which enters information regarding a found plug-in module into a
24 structure, the structure being used to create and to display a plug-in menu of
25 found plug-in modules, an "about" menu-create routine which enters "about"
plug-in information into an "about" structure, the "about" structure being used to
create and to display a plug-in "about" menu, a menu-enable routine, a menu-
disable routine and an invoking routine which permits the client application to
utilize a selected plug-in module.

26 Corrie, Jr., et al. is apparently directed to a "Method and system for
27 performing parametric testing of a functional programming interface" (the Title).

28 The Abstract of Corrie, Jr., et al. reads as follows:

1 The present invention provides a method and system for performing
2 parametric testing of a functional programming interface. Parametric testing of a
3 function verifies that the function performs as expected when a valid or an
4 invalid parameter is passed to the function. To perform parameter testing on a
5 function, the present invention receives as input prototype information for the
6 function, and then formulates a testing plan. The invention tests the function
7 according to the testing plan. The testing plan specifies a list of invalid and valid
8 values for each parameter of the function. The invention repeatedly invokes the
9 function, each time passing the function various combinations of invalid and
10 valid values. The function passes the test when (1) the function does not return
11 an error code for any combination of valid parameter values and (2) the function
12 returns an error for any combination of parameter values that include an invalid
13 parameter value.

14 None of Burger et al., Hayes, or Corrie, Jr. et al. are directed to *deriving* an
15 application programming interface (API). Furthermore, it is respectfully submitted
16 that none of Burger et al., Hayes, or Corrie, Jr. et al. describe, teach, or otherwise
17 render unpatentable **deriving the API from the core scenario responsive to the**
18 **plurality of code samples** (from example claim 1) in conjunction with the other
19 claimed elements.

20 The rejection of independent claim 1 from the current Office Action reads
21 on Page 10 as follows (with any highlighting removed):

22 As per Claim 1, Burger et al. disclose:

- 23 - preparing a plurality of code samples for a core scenario, each
24 respective code sample of the plurality of code samples corresponding to a
25 respective programming language of a plurality of programming languages (see
Column 5: 30-36, "... the invention generalizes existing entries to a plurality of
applications 16 written in any of a number of predetermined languages. "); and
- deriving the API from the core scenario responsive to the plurality of
code samples (see Column 5: 36-43, " ... the function of the generic APIs 14 of

1 the present invention is to transform parameters received in a number of
2 different formats determined by the particular language in which the application
3 16 is written into forms expected by the DBM APIs 12 and operating system
4 APIs 20 as dictated by the particular procedural language in which these APIs 12
5 and 20 are written. ").

6 In the current Office Action as reproduced above, the rejection of claim 1
7 cites to Column 5, Lines 30-43 of Burger et al. Burger et al. reads at Column 5,
8 Lines 13-43 as follows:

9 Still referring to FIG. 1, it will be recalled that one problem associated
10 with prior systems was that the operating system 18 and program services 10
11 assumed that application 16 would be written in the same language and
12 accordingly application program interfaces such as the APIs 20 to the operating
13 system 18 and the APIs to the particular program services 10 such as database
14 manager APIs 12 were interfaces written to a specific language. Thus, the APIs
15 12 and 20 were not generic and did not expect to be called from applications 16
16 written in more than one programming language. Thus, for example, if the
17 database manager program services 10 was written in the C-Language, the
18 associated DBM APIs 12 expected calls from the applications 16 such as a call
19 to create or delete a database to conform to calls in the C-Language and when
20 such calls from the applications 16 were in a different language, the applications
21 16 would not execute properly. In the present invention, however, a plurality of
22 generic APIs 14 are provided to these pre-existing entry points of the DBM APIs
23 12 and operating system APIs 20 whereby the invention generalizes existing
24 entries to a plurality of applications 16 written in any of a number of
25 predetermined languages. *Thus the function of the generic APIs 14 of the present
invention is to transform parameters received in a number of different formats
determined by the particular language in which the application 16 is written into
forms expected by the DBM APIs 12 and operating system APIs 20 as dictated
by the particular procedural language in which these APIs 12 and 20 are
written.*

(*italicized emphasis added by this Reply*)

23 As is apparent from the above excerpt of Burger et al., Burger et al. is
24 directed to a transformation of parameters by "generic APIs 14" so as to
25

1 accommodate the parameter formats “expected by the DBM APIs 12 and operating
2 system APIs 20”.

3 Hence, it is respectfully submitted that Burger et al. does not teach *deriving*
4 an API.

5
6 Moreover, it is respectfully submitted that Corrie, Jr., et al. is directed to
7 standard parameter testing on functions. It is not related to **performing one or**
8 **more usability studies on the API utilizing a plurality of developers** (e.g., from
9 dependent claim 5).

10
11
12 Thus, it is respectfully submitted that no art of record, either alone or in any
13 combination, anticipates or renders obvious the elements of independent claims 1,
14 15, 34, 47, 48, and 49. Moreover, although each of the pending dependent claims
15 also includes additional element(s) militating toward allowability, they are
16 allowable at least for the reasons given above in connection with the independent
17 claims.

18
19
20 Accordingly, it is respectfully requested that the §102 rejections and
21 §103(a) rejections be withdrawn.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

Respectfully Submitted,

By: Keith W. Saunders
Keith W. Saunders
Reg. No. 41,462
(509) 324-9256 x238